WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Term	Documents
(25 AND 14).USPT.	4
(L14 AND L25).USPT.	4

Database: US Patents Full-Text Database; US Pre-Grant Publication Full-Text Database; JPO Abstracts Database; EPO Abstracts Database; Derwent World Patents Index; IBM Technical Disclosure Bulletins

Search: L26

Refine Search

Recall Text | Clear

Search History

DATE: Sunday, April 21, 2002 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L26	l14 and l25	4	L26
L25	Java compiler$1	149	L25
L24	L23 and l19	13	L24
L23	exclusive near3 access$	1552	L23
L22	Java and l19	0	L22
L21	L20 and l19	0	L21
L20	JAVA	3621	L20
L19	modify$ near6 l14	31	L19
L18	l13 and l16 and l17	11	L18
L17	resource$1 near5 manag$	6431	L17
L16	performance goal$1	714	L16
L15	L14 and l12	11	L15
L14	shar$ near5 resource$1	6149	L14
L13	share$ near5 resource$1	5133	L13
L12	l1 and l11	41	L12
L11	per request	258	L11
L10	(L9 or l7) and l8	0	L10
L9	shareable resource$1	45	L9
L8	on demand	7	L8
L7	sharable resource$1	48	L7
L6	L5 and l4	1	L6
L5	banankhah.xp.	344	L5
L4	Eilert.in.	69	L4
L3	5675739.pn.	1	L3
L2	shar$	342404	L2
L1	resource$1 near5 access$	7397	L1

END OF SEARCH HISTORY

L24: Entry 1 of 13 File: USPT Mar 12, 2002

DOCUMENT-IDENTIFIER: US 6357016 B1
TITLE: Method and apparatus for disabling a clock signal within a multithreaded processor

Drawing Description Paragraph Right (16):
FIG. 13 is a flow chart illustrating a method, according to one embodiment, of providing exclusive access to an event handler within a multithreaded processor.

Drawing Description Paragraph Right (17):
FIG. 14 is a state diagram depicting operation, according to one embodiment, of an exclusive access state machine implemented within a multithreaded processor.

Detailed Description Paragraph Right (60):
The SYNCHRONIZATION event 262 is signaled by microcode when a particular thread (e.g., a first thread) is required to modify a shared state or resource within the multithreaded processor 30. To this end, the microcode sequencer 66 inserts a synchronization microinstruction into the flow for the first thread and, in order to avoid a deadlock situation, marks the "synchronization microinstruction" with both a shared resource flow marker 184 and a synchronization flow marker 186. The SYNCHRONIZATION event 262 is only detected (or registered) upon the retirement of the synchronization microinstruction for the first thread, and upon the retirement of a microinstruction for the second thread that has a synchronization flow marker 186 associated therewith. A SYNCHRONIZATION event 262 has the effect of invoking a synchronization event handler that restarts execution of the first thread at an instruction pointer stored in a microcode temporary register. Further details regarding the handling of a SYNCHRONIZATION event 262 are provided below. The second thread performs the virtual NUKE 260.

Detailed Description Paragraph Right (68):
If the first event for the first thread (e.g., thread 0) is determined not to modify a shared state or resource, the method 220 proceeds to decision box 284, where a determination is made as to whether the second thread (e.g., thread 1) is retiring a microinstruction that has a shared resource flow marker 184 associated therewith. Referring to FIG. 9, the retirement pointer 182 for the thread 1 is shown to reference a microinstruction having both a shared resource flow marker 184 and a synchronization flow marker 186. In this situation, the condition presented at decision box 284 will have been fulfilled, and the method 220 accordingly proceeds to block 280, where the multithreaded nuke operation is performed. Alternatively, should the retirement pointer 182 for the second thread (e.g., thread 1) not reference a microinstruction having either a shared resource flow marker 184 or a synchronization flow marker 186, the method proceeds to block 286, where retirement of the second thread continues by advancement of the retirement pointer 182. From the block 286, the method 220 loops back to the decision box 278, where a determination is again made whether the second thread has encountered an event.

Detailed Description Paragraph Right (92):
Certain event handlers (e.g., those for handling the paging and synchronization events,) require exclusive access to the multithreaded processor 30 to utilize shared resources and to modify shared state. Accordingly, the microcode sequencer 66 implements an exclusive access state machine 69 which gives exclusive access, in turn, to event handlers for the first and second threads where either of these event handlers requires such exclusive access. The exclusive access state machine 69 may only be referenced when more than one thread is active within the multithreaded processor 30. A flow marker, associated with an event handler that is provided with exclusive access, is inserted into the flow for the thread to mark the end of the exclusive code comprising the event handler. Once the exclusive access is completed

for all threads, the microcode sequencer 66 resumes normal issuance of microinstructions.

Detailed Description Paragraph Right (93):
FIG. 13 is a flowchart illustrating a method 400, according to exemplary embodiment, of providing exclusive access to an event handler 67 within a multithreaded processor 30. The method 400 commences at block 402 with the receipt by the microcode sequencer 66 of first and second event vectors, for respective first and second threads, from the event detector 188. As described above, each of the first and second event vectors will identify a respective event handler 67.

Detailed Description Paragraph Right (95):
At decision box 404, the microcode sequencer 66 makes a determination as to whether either of the first or second event handlers 67 requires exclusive access to a shared resource, or modifies a shared state. If so, at block 406 the microcode sequencer 66 implements the exclusive access state machine 69 to provide exclusive access, in turn, to each of the first and second event handlers 67. FIG. 14 is a state diagram depicting operation, according to exemplary embodiment, of the exclusive access state machine 69. The state machine 69 is shown to include five states. In a first state 408, microcode for the first and second threads is both issued by the microcode sequencer 66. On the occurrence of a nuke operation 410 responsive to an event that requires an exclusive access event handler, the state machine 69 transitions to a second state 412, wherein a first event handler 67 (i.e., microinstructions), associated with an event for a first thread, is issued. Following the sequencing of all microinstructions that constitute the first event handler 67, and also following completion of all operations instructed by such microinstructions, the microcode sequencer 66 then issues a stall microinstruction (e.g., microinstruction having an associated stall flow marker) at 414 to transition the state machine 69 from the second state 412 to a third state 416 in which issuance of a first thread microinstructions is stalled. At 418, the stall microinstruction issued at 414 is retired from the reorder buffer 162 to thereby transition the state machine 69 from the third state 416 to a fourth state 420 in which the microcode sequencer 66 issues the second event handler 67, associated with an event for the second thread. Following the sequencing of all microinstructions that constitute the second event handler 67, and also following the completion of all operations instructed by such microinstructions, the microcode sequencer 66 then issues a further stall microinstruction at 422 to transition the state machine 69 from the fourth state to a fifth state 424 in which the second event handler 67 is stalled. At 426, the stall microinstruction issued at 422 is retired from the reorder buffer 162 to thereby transition the state machine 69 from the fifth state 424 back to the first state 408.

Detailed Description Paragraph Right (97):
Alternatively, if it is determined the decision box 404 that neither of the first or second event handlers require exclusive access to shared resources or state of the processor 30, the method proceeds to block 434, where the microcode sequencer 66 sequences microcode constituting the first and second event handlers 67 a non-exclusive, interleaved manner.

Detailed Description Paragraph Center (10):
Exclusive Access by an Event Handler

WEST

End of Result Set

☐ Generate Collection Print

L6: Entry 1 of 1 File: USPT Aug 28, 2001

US-PAT-NO: 6282560
DOCUMENT-IDENTIFIER: US 6282560 B1

TITLE: Managing processor resources in a non-dedicated computer system

DATE-ISSUED: August 28, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Eilert; Catherine Krueger	Wappingers Falls	NY		
Yocom; Peter Bergersen	Wappingers Falls	NY		

US-CL-CURRENT: 709/100; 709/102

CLAIMS:

What is claimed is:

1. A method of managing processor resources in a general purpose computer system, comprising:

allocating an amount of a processor resource to a real-time application of said general purpose computer system, said amount not to exceed a limit chosen for a group of one or more real-time applications, said group including at least said real-time application, wherein a selected amount of said processor resource remains available for at least one non-real-time application of said general purpose computer system, and wherein said processor resource comprises processing capacity of one or more central processing units; and

processing said real-time application, wherein said processing comprises preventing, by a resource manager without assistance from said real-time application, said real-time application from exceeding a maximum amount of said processor resource selected for said group of real-time applications.

2. The method of claim 1, further comprising indicating to said computer system that said real-time application wishes to be processed.

3. The method of claim 2, wherein said indicating comprises:

setting a dispatch priority chosen for said real-time application; and

specifying said dispatch priority is not to be adjusted.

4. The method of claim 1, wherein said allocating comprises:

determining a highest service cost, within a given period of time, for delivering a real-time data rate;

calculating a service rate for delivering a real-time data stream of said real-time application; and

indicating a sufficient amount of said processor resource is available for said real-time data stream when said service rate falls within said limit.

5. The method of claim 4, further comprising adding a data rate of said real-time

data stream to a current allocated data rate in order to track the amount of processor resource currently allocated to said group of real-time applications.

6. The method of claim 1, further comprising deallocating said allocated amount of said processor resource when said real-time application no longer needs said processor resource.

7. The method of claim 6, wherein said deallocating comprises subtracting a data rate of said real-time application from a current allocated data rate such that the deallocated amount of processor resource is now available for said group of real-time applications.

8. The method of claim 1, further comprising pausing processing of said real-time application.

9. The method of claim 8, wherein said pausing comprises reserving said allocated amount of processor resource.

10. The method of claim 8, further comprising resuming processing of said paused real-time application.

11. The method of claim 1, wherein said processing of said real-time application comprises adjusting one or more control parameters associated with said real-time application, when said one or more control parameters need adjusting to meet one or more performance goals of said computer system.

12. The method of claim 1, further comprising processing one of said at least one non-real-time application, wherein said processing of said one non-real-time application comprises adjusting one or more control parameters associated with said one non-real-time application, when said one or more control parameters need adjusting to meet one or more performance goals of said computer system.

13. The method of claim 1, wherein said preventing comprises indicating one or more work units of said real-time application is non-dispatchable such that said maximum amount is not exceeded.

14. A method of managing processor resources in a computer system, comprising:

selecting an amount of a processor resource allocatable to a group of one or more real-time applications of said computer system, said processor resource comprising processing capacity of one or more central processing units;

reserving a chosen amount of said processor resource for one or more non-real-time applications of said computer system; and

preventing, by a resource manager without assistance of said one or more real-time applications, said group of one or more real-time applications from exceeding a maximum amount of said processor resource selected for said group of one or more real-time applications.

15. The method of claim 14, wherein said selecting comprises selecting an amount of said processor resource allocatable to a plurality of groups of one or more real-time applications.

16. An article of manufacture comprising:

a computer useable medium having computer readable program code means embodied therein for causing the managing of processor resources in a general purpose computer system, the computer readable program code means in said article of manufacture comprising:

computer readable program code means for causing a computer to effect allocating an amount of a processor resource to a real-time application of said general purpose computer system, said amount not to exceed a limit chosen for a group of one or more real-time applications, said group including at least said real-time application, wherein a selected amount of said processor resource remains available for at least one non-real-time application of said general purpose computer system, and wherein said processor resource comprises processing capacity of one or more central processing units; and

computer readable program code means for causing a computer to effect processing said real-time application, wherein said computer readable program code means for causing a computer to effect processing comprises computer readable program code means for causing a computer to effect preventing, by a resource manager without assistance from said real-time application, said real-time application from exceeding a maximum amount of said processor resource selected for said group of real-time applications.

17. The article of manufacture of claim 16, further comprising computer readable program code means for causing a computer to effect indicating to said computer system that said real-time application wishes to be processed.

18. The article of manufacture of claim 17, wherein said computer readable program code means for causing a computer to effect indicating comprises:

computer readable program code means for causing a computer to effect setting a dispatch priority chosen for said real-time application; and

computer readable program code means for causing a computer to effect specifying said dispatch priority is not to be adjusted.

19. The article of manufacture of claim 16, wherein said computer readable program code means for causing a computer to effect allocating comprises:

computer readable program code means for causing a computer to effect determining a highest service cost, within a given period of time, for delivering a real-time data rate;

computer readable program code means for causing a computer to effect calculating a service rate for delivering a real-time data stream of said real-time application; and

computer readable program code means for causing a computer to effect indicating a sufficient amount of said processor resource is available for said real-time data stream when said service rate falls within said limit.

20. The article of manufacture of claim 19, further comprising computer readable program code means for causing a computer to effect adding a data rate of said real-time data stream to a current allocated data rate in order to track the amount of processor resource currently allocated to said group of real-time applications.

21. The article of manufacture of claim 16, further comprising computer readable program code means for causing a computer to effect deallocating said allocated amount of said processor resource when said real-time application no longer needs said processor resource.

22. The article of manufacture of claim 21, wherein said computer readable program code means for causing a computer to effect deallocating further comprises computer readable program code means for causing a computer to effect subtracting a data rate of said real-time application from a current allocated data rate such that the deallocated amount of processor resource is now available for said group of real-time applications.

23. The article of manufacture of claim 16, further comprising computer readable program code means for causing a computer to effect pausing processing of said real-time application.

24. The article of manufacture of claim 23, wherein said computer readable program code means for causing a computer to effect pausing comprises computer readable program code means for causing a computer to effect reserving said allocated amount of processor resource.

25. The article of manufacture of claim 23, further comprising computer readable program code means for causing a computer to effect resuming processing of said paused real-time application.

26. The article of manufacture of claim 16, wherein said computer readable program code means for causing a computer to effect processing of said real-time application

comprises computer readable program code means for causing a computer to effect adjusting one or more control parameters associated with said real-time application, when said one or more control parameters need adjusting to meet one or more performance goals of said computer system.

27. The article of manufacture of claim 16, further comprising computer readable program code means for causing a computer to effect processing one of said at least one non-real-time application, wherein said processing of said one non-real-time application comprises adjusting one or more control parameters associated with said one non-real-time application, when said one or more control parameters need adjusting to meet one or more performance goals of said computer system.

28. The article of manufacture of claim 16, wherein said computer readable program code means for causing a computer to effect preventing comprises computer readable program code means for causing a computer to effect indicating one or more work units of said real-time application is non-dispatchable such that said maximum amount is not exceeded.

29. An article of manufacture comprising:

a computer useable medium having computer readable program code means embodied therein for causing the managing of processor resources in a computer system, the computer readable program code means in said article of manufacture comprising:

computer readable program code means for causing a computer to effect selecting an amount of a processor resource allocatable to a group of one or more real-time applications of said computer system, said processor resource comprising processing capacity of one or more central processing units;

computer readable program code means for causing a computer to effect reserving a chosen amount of said processor resource for one or more non-real-time applications of said computer system; and

computer readable program code means for causing a computer to effect preventing, by a resource manager without assistance of said one or more real-time applications, said group of one or more real-time applications from exceeding a maximum amount of said processor resource selected for said group of one or more real-time applications.

30. The article of manufacture of claim 29, wherein said computer readable program code means for causing a computer to effect selecting comprises computer readable program code means for causing a computer to effect selecting an amount of said processor resource allocatable to a plurality of groups of one or more real-time applications.

31. The method of claim 1, wherein said real-time application is prevented from overrunning said computer system.

32. The method of claim 16, wherein said real-time application is prevented from overrunning said computer system.